UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/706,867	11/07/2000	Edward Colles Nevill	P007773US	8402

7590 10/31/2003

Nixon & Vanderhye P C
1100 North Glebe Road
8th Floor
Arlington, VA 22201-4714

EXAMINER	
STEELMAN, MARY J	
ART UNIT	**PAPER NUMBER**
2122	4

DATE MAILED: 10/31/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/706,867	NEVILL, EDWARD COLLES	
	Examiner	Art Unit	
	Mary J. Steelman	2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11/08/00,02/01/01,01/16/02* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-19* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on *07 November 2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☒ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. Claims 1-19 are pending.

Priority

2. Acknowledgment is made of applicant's claim for foreign priority based on an application filed in the United Kingdom (001030.6) on 01/17/2000. It is noted, however, that applicant has not filed a certified copy of the UK Patent Application as required by 35 U.S.C. 119(b).

Specification

3. Applicant is reminded to provide Cross-References to Related Applications (See 37 CFR 1.78 and MPEP § 201.11.)

4. The disclosure is objected to because of the following informalities:

Page 5, lines 25-29 repeat the previous paragraph. Delete lines 25-29.

Appropriate correction is required.

Claim Objections

5. Claims 17 and 18 are objected to because of the following informalities:

Claim 17, page 19, lines 1 and 2, and claim 18, page 19, lines 6 and 7, recite, "...code instructions words..." should be –code instruction words--. Delete the 's' from instructions.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Regarding claims 8 and 9, the phrase "substantially" renders the claims indefinite because it is unclear whether the limitation(s) following the phrase are part of the claimed invention. See

MPEP § 2173.05(d). Claim 8 will be treated as if it states, "...wherein a plurality of interpreted code instruction words correspond to a native code instruction words."

Regarding claim 16, the phrase "...software interpreter is stored within said cache memory both when executing native program instruction words and interpreted program instruction words." It would be clearer if the phrase recited, "...software interpreter is stored within said cache memory when both native program instruction words and interpreted program instruction words are executing." The former phrase implies that the interpreter executes native program instruction words.

Claim Rejections - 35 USC § 102

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

8. Claims 1 – 5, 8, and 19 are rejected under 35 U.S.C. 102(e) as being anticipated by US Patent 6,513,156 to Bak.

Per claims 1 and 19:

-a processing unit responsive to native program instruction words to perform data processing operations; (Fig. 5. Col. 8, lines 1-2, "hybrid virtual and native machine instructions...")

-an instruction interpreter responsive to one or more interpreted instruction words specifying a data processing operation to execute native instruction words upon said processing unit to perform said data processing operation; (Col. 8, lines 7-10, "The interpreter generates modified JAVA virtual machine instructions by overwriting bytecode with a go_native virtual machine instruction.")

-a memory for storing said computer program wherein... (Fig. 2, and col. 4, lines 47-50, "...system memory...to store and retrieve software...that implements the invention...")

-said computer program includes both native instruction words and interpreted instruction words; (Fig. 5.)

-a native code portion invokes interpretation of an interpreted code portion by executing a native code call instruction to said instruction interpreter; -execution of said native code call instruction triggers generation of a return address specifying a location within said memory for said native code call instruction; -said instruction interpreter uses said return address as a pointer to said interpreted code portion within said memory. (Fig. 11 and col. 12, lines 13-17, "Once the virtual function finishes execution (native code finishes), the system returns to the return address that was pushed on the stack at step 907. At the return address, there are native machine instructions for the interpreter to reload the saved bytecode pointer...")

Per claim 2:

-return address is an address immediately succeeding an address of said native code call instruction within said memory. (See fig. 5, and col. 8, lines 36-39, "The native machine instructions in the snippet perform the same operations as if the bytecodes 2-5 would have been interpreted. Afterwards, the interpreter continues with the execution of bytecode 6 as if no snippet existed.")

Per claim 3:

-instruction interpreter writes a new value of said return address for use by a native code return instruction as a pointer to a next native code instruction to be executed when interpretation of said interpreted code portion is finished and return is being made to execution of said next native

code portion of said computer program. (When snippet finishes and returns to interpreter, the interpreter may have a successive go_native instruction pointing to a second snippet. See figure 13. Any number of native code snippets may be stored. A virtual machine go_native instruction references the snippets.)

Per claim 4:

-interpreted code portion finishes with an interpreted return instruction that returns execution to a native code portion at an address pointed to by said return address. (Col. 8, lines 32-35, "When the interpreter executes the go_native bytecode, the interpreter will look up the snippet in the snippet zone specified by the go_native bytecode and then activate the native machine instructions in the snippet.")

Per claim 5:

-interpreted code portion finishes with an interpreted exit instruction that results in execution of a next native code portion stored within said memory at an address immediately following said exit instruction without a return being made corresponding to said native code call instruction. (See figure 13, interpreter code followed by exit instruction (go_native instruction), results in execution of native code.)

Per claim 8:

-interpreted code instruction word corresponds to a native code instruction word. (See fig. 5 and col. 3, lines 5-8, "The new virtual machine instruction may include a pointer to a data block in which is stored the native machine instructions, the copy of the selected virtual machine instruction (correspond)...")

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 6, 7, and 10-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,513,156 to Bak et al., in view of US Patent 6,081,665 to Nilsen et al.

Per claims 6 and 7:

Bak disclosed the use of a hybrid system that processed both bytecode and native machine instructions for the purpose of increasing execution efficiency. Bak disclosed (col. 12, lines 6-22), "The system pushes the interpreter return address on the stack...The interpreter return address is a predefined location where the execution of the interpreter...should resume. The native machine instructions...instruct the system to jump to the function specified...Once the virtual function (the native instructions) finishes...the system returns to the return address that was pushed on the stack...At the return address, there are native machine instructions for the interpreter to reload the saved bytecode pointer...so the interpreter may continue where it left off..." Bak failed to discuss the register storage as used by executing native instructions. However, Nilsen disclosed a "Processing unit includes a bank of data processing registers and said return address is stored within a predetermined data processing register within said bank of data processing registers upon execution of said native code call instruction." And "-memory includes a stack memory region and said return address is copied from said predetermined data processing register to said stack memory region upon invocation of said interpreted code portion."

(See fig. 94, saved registers. Also, col. 17, line 44 describes the processes that are implemented when bytecode is called. Col. 18, line 45, describes the processes that are implemented when native instructions are called. "Copies the register-held psp and npsp registers into global memory locations...Then assigns sp and fp (the machine's stack and frame pointer registers) to reflect the current c-stack context...Copies the return address of the non-pointer stack and saves its value...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Bak's disclosed invention that processes native and bytecode instructions, by including information regarding the register storage as well as the stack storage, because it is well known that a virtual machine is a stack machine, storing needed values on the stack, and it is well known that a processor for native instructions frequently has a register based architecture, and as processing switches from one instruction set to the other, some technique is necessary to copy values from the register based machine to the stack based machine and vice-versus.

Per claims 10-13:

Bak disclosed the use of a hybrid system that processed both bytecode and native machine instructions. Bak failed to discuss the bit size of native code instruction words and interpreted code instruction words and the bit size of the data processing operations to be performed.

However, Nilsen disclosed, "Byte code is a term of art that describes a method of encoding instructions (for interpretation by a virtual machine) as 8-bit numbers, each pattern of 8 bits representing a different instruction" (col. 3, lines 29-32). Also, col. 12, lines 23-24,

"The...virtual machine consists primarily of an interpreter for the ...byte-code instruction set..." and lines 40-50, ""The...programmer can choose to implement certain methods in C (frequently 32 bit instructions). At run-time, these methods are represented by native machine code...All other PERC methods are written in PERC. At runtime, certain methods written in PERC are represented as PERC byte codes (8 bit instructions)..." Additionally, Nilsen disclosed that both native code instruction words and interpreted code instruction words specify N-bit data processing operations to be performed, as an example (col. 8, lines 45-64, "...the signature structure used to represent the memory layout of heap-allocated objects...When the garbage collector (used with bytecode execution) scans the corresponding object in search of pointers, it looks no further than the word numbered last_descriptor.type_code comprises a 2 bit type tag...with the remaining 30 bits representing the value..." Furthermore, ultimately all byte code instructions (8 bit) map to native instructions (could be 32 bits) for processing.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have considered that native code instruction words (which can have 32 bits) have more bits than interpreted code words (bytecodes of 8 bits) and that a 32 bit architecture processing bytecode, using a virtual machine, ultimately maps each processing operation to the 32 bit architecture. Virtual machines processing 8-bit bytecode hosted on a 32-bit processor are well known in the art, and thus inherent.

11. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,513,156 to Bak et al.

Per claim 9, "native code instruction words form a native code instruction set and said interpreted code instruction words form an interpreted code instruction set, data processing

operations provided by said interpreted code instruction set being a subset of data processing operations provided by said native code instruction set."

It is well known that a virtual machine / interpreter hosted on a native machine interprets instructions that are ultimately mapped to machine code of the host machine. Therefore, it is inherent that the virtual machine / interpreter uses a subset of instructions of the host machine instruction set.

12. Claims 14-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,513,156 to Bak et al., in view of US Patent Application 2003/191792 to Waki et al. (April 1999).

Per claims 14-16, Bak disclosed the use of a hybrid system that processed both bytecode and native machine instructions. Bak failed to disclose that the interpreter is formed of native code instruction words, and can be stored in cache memory for the duration of executing native program instruction words and interpreted program instruction words.

However, Waki disclosed, at page 13, paragraph [0248], "The virtual machine (an interpreter) and the virtual machine compiler of the present embodiment are programs written with instructions for the real machine (native code)." Per claims 15 and 16, Waki disclosed, "cache memory and wherein said software interpreter is, in use, stored within said cache memory." At page 5, paragraph [0078], "...sample virtual machine program shown in Fig. 27 is stored in the cache memory..." Also, page 6, paragraph [oo85], "...a high speed virtual machine system that can be used by a real machine with a cache system..."

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have included details on the interpreter formed of native code instruction words

and allow for the possibility of storing the interpreter in cache because the host machine must be able to process the interpreted instructions therefore they must be in the host machine language and an interpreter, a running program, located in a cache, provides a high speed memory location, allowing for more efficient execution.

Per claims 17 and 18, Bak disclosed the use of a hybrid system that processed both bytecode and native machine instructions. Bak failed to disclose information regarding execution frequency and speed in making choices between using compiled native code versus using interpreted bytecode.

However, Waki disclosed execution speed and execution frequency are considerations when deciding which parts of the code to leave as bytecode to be interpreted (slower) and which parts of the code to process as native code (faster) at page 2, paragraph [0021], "...how to increase execution speed..." and at page 3, paragraph [0026], "...execution speed being increased in proportion to the reduction in the number of memory accesses (frequency)..." Page 3, paragraph [0028] discusses a "...predetermined part of the virtual machine program is written in real machine instructions..." showing that some consideration is given to which parts will execute more efficiently in native code. At page 7, paragraph [0096], "...replace...stack operations (used in virtual machine byte code interpreting)...with read/write operations for the internal registers (used in native code processing) ...execution efficiency of the virtual machine is raised." At page 8, paragraph [0114], "...control switches between executing a virtual machine function and a real machine function...improved execution speed..." Page 10, paragraph [0133], "...the present invention improves execution speed...promote efficient and

high-speed use of shared resources..." Additionally, Waki considers the frequency on interrupt handling (page 8, paragraph [0108]) to avoid decreases in execution speed.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have included information regarding improved efficiency, considering execution speed and execution frequency, because this is the purpose of modifying byte code to native code.

Conclusion

13. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

US Patent 6,151,703 to Crelier (See column 10, program execution calls from interpreted code as well as compiled code.)

US Patent 6,298,434 to Lindwer (Preprocessor fetches virtual machine instructions & generates native instructions.)

US Patent 6,564, 241 to Rosengard (Interpreter stored within cache memory.)

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (703) 305-4564. The examiner can normally be reached Monday through Thursday, from 7:00 A.M. to 5:30 P.M. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on (703) 305-4552.

The fax phone numbers are (703) 872-9306 for regular communications and for After Final communications. Any inquiry of a general nature or relating to the status of this

application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Mary Steelman

10/22/2003

ANTONY NGUYEN-BA
PRIMARY EXAMINER